

CLAIMS

What is claimed is:

1. A method of content addressable data storage and compression for semi-persistent computer memory comprising:
 - 5 providing a chunk of data comprising a quantity of input data;
 - retrieving a memory block from semi-persistent computer memory;
 - searching for a segment of the chunk that matches the memory block; and
 - 10 if a matching segment is found:
 - discarding the matching segment;
 - 15 providing a retrieval key for the memory block as a retrieval key for the matching segment;
 - identifying an unmatched portion of the chunk that does not match the memory block;
 - 20 identifying a free memory block of a file system;
 - storing the unmatched portion semi-persistently in the free memory block; and
 - 25 providing a retrieval key for the unmatched portion.

2. The method of claim 1 wherein a free memory block of a file system has a block size at least as large as a maximum memory block size.
3. The method of claim 1 wherein storing the unmatched portion semi-persistently in the free memory block comprises storing the unmatched portion without recording the use of the free memory block in the file system.

4. The method of claim 1 wherein:

identifying a free memory block of a file system comprises reading a block identification from a free block list of a file system, and

5

storing the unmatched portion semi-persistently in the free memory block comprises leaving the block identification unchanged in the free block list of the file system.

5. The method of claim 1 wherein searching for a segment of the chunk that matches the memory block comprises searching at a repeating memory interval through a search section of the chunk for a segment of the chunk that matches the memory block.

5

6. The method of claim 5 wherein searching at a repeating memory interval through a search section of the chunk for a segment of the chunk that matches the memory block comprises:

5

calculating a weak checksum for the memory block;

calculating weak checksums for segments of the search section of the chunk;

10 comparing the weak checksums for the segments with the checksum for the
memory block; and

if a segment is found with a weak checksum equal to the weak checksum of
the memory block:

15 calculating a strong checksum for the memory block;

calculating a strong checksum for the segment with the matching weak
checksum;

20 comparing the strong checksum of the memory block and the strong
checksum for the segment with the equal weak checksum; and

25 determining that the search has found a segment having contents that match
the contents of the memory block if the strong checksum of the memory block
and the strong checksum for the segment with the matching weak checksum
are equal.

7. The method of claim 1 wherein storing the unmatched portion of the chunk
comprises storing the unmatched portion of the chunk as a new memory block
having a memory block size equal to the size of the unmatched portion of the
chunk.

5

8. The method of claim 1 wherein searching for a segment of the chunk that
matches the memory block fails to find a matching segment, the method
further comprising repeatedly carrying out the following steps for all memory
blocks in computer memory until a matching segment is found:

5

retrieving a next memory block from computer memory; and

searching for a segment of the chunk that matches the next memory block.

9. The method of claim 8 wherein no matching segment is found in any memory block in computer memory, the method further comprising:

identifying a free memory block of a file system;

5

storing a search section of the chunk semi-persistently in the free memory block; and

providing a retrieval key for the search section of the chunk.

10

10. The method of claim 8 wherein storing a search section of the chunk comprises storing the search section of the chunk as a new memory block having a memory block size equal to the size of the search section of the chunk.

5

11. The method of claim 8 wherein providing a retrieval key for a search section of a chunk comprises:

calculating a weak checksum for the search section of the chunk; and

5

calculating a strong checksum for the search section of the chunk.

12. The method of claim 1 further comprising:

receiving a retrieval key;

5 identifying a memory block in dependence upon the retrieval key;

retrieving the identified memory block; and

verifying the contents of the memory block.

10

13. The method of claim 12 wherein the retrieval key for the memory block comprises a unique key calculated with an algorithm that generates a unique key from the contents of a memory block, and verifying the contents of the memory block further comprises:

5

calculating a new key for the memory block with the same algorithm; and

comparing the retrieval key and the new key.

14. A system for content addressable data storage and compression for semi-persistent computer memory comprising:

means for providing a chunk of data comprising a quantity of input data;

5

means for retrieving a memory block from semi-persistent computer memory;

means for searching for a segment of the chunk that matches the memory block;

10

means for discarding a matching segment;

means for providing a retrieval key for the memory block as a retrieval key for the matching segment;

15

means for identifying an unmatched portion of the chunk that does not match the memory block;

means for identifying a free memory block of a file system;

20

means for storing the unmatched portion semi-persistently in the free memory block; and

means for providing a retrieval key for the unmatched portion.

25

15. The system of claim 14 wherein a free memory block of a file system has a block size at least as large as a maximum memory block size.

16. The system of claim 14 wherein means for storing the unmatched portion

semi-persistently in the free memory block comprises means for storing the unmatched portion without recording the use of the free memory block in the file system.

5

17. The system of claim 14 wherein:

means for identifying a free memory block of a file system comprises means for reading a block identification from a free block list of a file system, and

5

means for storing the unmatched portion semi-persistently in the free memory block comprises means for leaving the block identification unchanged in the free block list of the file system.

18. The system of claim 14 wherein means for searching for a segment of the chunk that matches the memory block comprises means for searching at a repeating memory interval through a search section of the chunk for a segment of the chunk that matches the memory block.

5

19. The system of claim 18 wherein means for searching at a repeating memory interval through a search section of the chunk for a segment of the chunk that matches the memory block comprises:

5

means for calculating a weak checksum for the memory block;

means for calculating weak checksums for segments of the search section of the chunk;

10

means for comparing the weak checksums for the segments with the checksum for the memory block;

means for calculating a strong checksum for the memory block;

15 means for calculating a strong checksum for the segment with the matching weak checksum;

means for comparing the strong checksum of the memory block and the strong checksum for the segment with the equal weak checksum; and

20

means for determining that the search has found a segment having contents that match the contents of the memory block if the strong checksum of the memory block and the strong checksum for the segment with the matching weak checksum are equal.

25

20. The system of claim 14 wherein means for storing the unmatched portion of the chunk comprises means for storing the unmatched portion of the chunk as a new memory block having a memory block size equal to the size of the unmatched portion of the chunk.

5

21. The system of claim 14 wherein means for searching for a segment of the chunk that matches the memory block fails to find a matching segment, the system further comprising means for repeatedly carrying out the following steps for all memory blocks in computer memory until a matching segment is found:

5

retrieving a next memory block from computer memory; and

searching for a segment of the chunk that matches the next memory block.

10

22. The system of claim 21 further comprising:

means for identifying a free memory block of a file system;

5 means for storing a search section of the chunk semi-persistently in the free memory block; and

means for providing a retrieval key for the search section of the chunk.

23. The system of claim 21 wherein means for storing a search section of the chunk comprises means for storing the search section of the chunk as a new memory block having a memory block size equal to the size of the search section of the chunk.

5

24. The system of claim 21 wherein means for providing a retrieval key for a search section of a chunk comprises:

means for calculating a weak checksum for the search section of the chunk;

5

and

means for calculating a strong checksum for the search section of the chunk.

25. The system of claim 14 further comprising:

means for receiving a retrieval key;

5 means for identifying a memory block in dependence upon the retrieval key;

means for retrieving the identified memory block; and

means for verifying the contents of the memory block.

10

26. The system of claim 25 wherein the retrieval key for the memory block comprises a unique key calculated with an algorithm that generates a unique key from the contents of a memory block, and means for verifying the contents of the memory block further comprises:

15

means for calculating a new key for the memory block with the same algorithm; and

means for comparing the retrieval key and the new key.

27. A computer program product for content addressable data storage and compression for semi-persistent computer memory comprising:

a recording medium;

5

means, recorded on the recording medium, for providing a chunk of data comprising a quantity of input data;

10

means, recorded on the recording medium, for retrieving a memory block from semi-persistent computer memory;

means, recorded on the recording medium, for searching for a segment of the chunk that matches the memory block;

15

means, recorded on the recording medium, for discarding a matching segment;

means, recorded on the recording medium, for providing a retrieval key for the memory block as a retrieval key for the matching segment;

20

means, recorded on the recording medium, for identifying an unmatched portion of the chunk that does not match the memory block;

means, recorded on the recording medium, for identifying a free memory block of a file computer program product;

25

means, recorded on the recording medium, for storing the unmatched portion semi-persistently in the free memory block; and

means, recorded on the recording medium, for providing a retrieval key for

30 the unmatched portion.

28. The computer program product of claim 27 wherein a free memory block of a file computer program product has a block size at least as large as a maximum memory block size.

29. The computer program product of claim 27 wherein means, recorded on the recording medium, for storing the unmatched portion semi-persistently in the free memory block comprises means, recorded on the recording medium, for storing the unmatched portion without recording the use of the free memory
5 block in the file computer program product.

30. The computer program product of claim 27 wherein:

means, recorded on the recording medium, for identifying a free memory block of a file computer program product comprises means, recorded on the
5 recording medium, for reading a block identification from a free block list of a file computer program product, and

means, recorded on the recording medium, for storing the unmatched portion semi-persistently in the free memory block comprises means, recorded on the
10 recording medium, for leaving the block identification unchanged in the free block list of the file computer program product.

31. The computer program product of claim 27 wherein means, recorded on the recording medium, for searching for a segment of the chunk that matches the memory block comprises means, recorded on the recording medium, for searching at a repeating memory interval through a search section of the
5 chunk for a segment of the chunk that matches the memory block.

32. The computer program product of claim 31 wherein means, recorded on the recording medium, for searching at a repeating memory interval through a search section of the chunk for a segment of the chunk that matches the memory block comprises:

5

means, recorded on the recording medium, for calculating a weak checksum for the memory block;

10

means, recorded on the recording medium, for calculating weak checksums for segments of the search section of the chunk;

means, recorded on the recording medium, for comparing the weak checksums for the segments with the checksum for the memory block;

15

means, recorded on the recording medium, for calculating a strong checksum for the memory block;

means, recorded on the recording medium, for calculating a strong checksum for the segment with the matching weak checksum;

20

means, recorded on the recording medium, for comparing the strong checksum of the memory block and the strong checksum for the segment with the equal weak checksum; and

25

means, recorded on the recording medium, for determining that the search has found a segment having contents that match the contents of the memory block if the strong checksum of the memory block and the strong checksum for the segment with the matching weak checksum are equal.

33. The computer program product of claim 27 wherein means, recorded on the recording medium, for storing the unmatched portion of the chunk comprises means, recorded on the recording medium, for storing the unmatched portion of the chunk as a new memory block having a memory block size equal to the size of the unmatched portion of the chunk.

34. The computer program product of claim 27 wherein means, recorded on the recording medium, for searching for a segment of the chunk that matches the memory block fails to find a matching segment, the computer program product further comprising means, recorded on the recording medium, for repeatedly carrying out the following steps for all memory blocks in computer memory until a matching segment is found:

retrieving a next memory block from computer memory; and

searching for a segment of the chunk that matches the next memory block.

35. The computer program product of claim 34 further comprising:

means, recorded on the recording medium, for identifying a free memory block of a file computer program product;

means, recorded on the recording medium, for storing a search section of the chunk semi-persistently in the free memory block; and

means, recorded on the recording medium, for providing a retrieval key for the search section of the chunk.

36. The computer program product of claim 34 wherein means, recorded on the recording medium, for storing a search section of the chunk comprises means, recorded on the recording medium, for storing the search section of the chunk as a new memory block having a memory block size equal to the size of the search section of the chunk.

37. The computer program product of claim 34 wherein means, recorded on the recording medium, for providing a retrieval key for a search section of a chunk comprises:

means, recorded on the recording medium, for calculating a weak checksum for the search section of the chunk; and

means, recorded on the recording medium, for calculating a strong checksum for the search section of the chunk.

10

38. The computer program product of claim 27 further comprising:

means, recorded on the recording medium, for receiving a retrieval key;

means, recorded on the recording medium, for identifying a memory block in dependence upon the retrieval key;

means, recorded on the recording medium, for retrieving the identified memory block; and

10

means, recorded on the recording medium, for verifying the contents of the memory block.

39. The computer program product of claim 38 wherein the retrieval key for the memory block comprises a unique key calculated with an algorithm that generates a unique key from the contents of a memory block, and means, recorded on the recording medium, for verifying the contents of the memory block further comprises:
- 5
- means, recorded on the recording medium, for calculating a new key for the memory block with the same algorithm; and
- 10
- means, recorded on the recording medium, for comparing the retrieval key and the new key.